2012 IEEE 53rd Annual Symposium on Foundations of Computer Science

The Locality of Distributed Symmetry Breaking**

Leonid Barenboim*, Michael Elkin* , Seth Pettie†, and Johannes Schneider ‡
*Department of Computer Science
Ben-Gurion University of the Negev, Beer-Sheva, Israel.
Email: {leonidba, elkinm}@cs.bgu.ac.il
†Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, MI, USA.
Email: pettie@umich.edu
‡Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
Email: schneider@tik.ee.ethz.ch

***Abstract*—We present new bounds on the locality of several classical symmetry breaking tasks in distributed networks. A sampling of the results include**

1) **A randomized algorithm for computing a maximal matching (MM) in $O(\log\Delta+(\log\log n)^4)$ rounds, where Δ is the maximum degree. This improves a 25-year old randomized algorithm of Israeli and Itai that takes $O(\log n)$ rounds and is *provably optimal* for all $\log\Delta$ in the range $[(\log\log n)^4, \sqrt{\log n}]$.**
2) **A randomized maximal independent set (MIS) algorithm requiring $O(\log\Delta\sqrt{\log n})$ rounds, for all Δ, and only $2^{O(\sqrt{\log\log n})}$ rounds when $\Delta = \mathrm{poly}(\log n)$. These improve on the 25-year old $O(\log n)$-round randomized MIS algorithms of Luby and Alon, Babai, and Itai when $\log\Delta \ll \sqrt{\log n}$.**
3) **A randomized $(\Delta+1)$-coloring algorithm requiring $O(\log\Delta + 2^{O(\sqrt{\log\log n})})$ rounds, improving on an algorithm of Schneider and Wattenhofer that takes $O(\log\Delta+\sqrt{\log n})$ rounds. This result implies that an $O(\Delta)$-coloring can be computed in $2^{O(\sqrt{\log\log n})}$ rounds for all Δ, improving on Kothapalli et al.'s $O(\sqrt{\log n})$-round algorithm.**

We also introduce a new technique for reducing symmetry breaking problems on low arboricity graphs to low degree graphs. Corollaries of this reduction include MM and MIS algorithms for low arboricity graphs (e.g., planar graphs and graphs that exclude any fixed minor) requiring $O(\sqrt{\log n})$ and $O(\log^{2/3} n)$ rounds w.h.p., respectively.

***Keywords*-Coloring; Maximal Independent Set; Maximal Matching;**

I. Introduction

Breaking symmetry is one of the central themes in the theory of distributed computation. At initialization the nodes of a distributed system are assumed to be in the same state (but with distinct node IDs), yet to perform any computation the nodes frequently must take different roles, that is, they must somehow break their initial symmetry. In this paper we study three of the classical symmetry breaking tasks in Linial's $\mathcal{LOCAL}$ model [20]: computing *maximal matchings* (MM), *maximal independent sets* (MIS), and $(\Delta+1)$-coloring, where Δ is the maximum degree.[1] In the $\mathcal{LOCAL}$ model each node of the input graph G hosts a processor, which is aware of its neighbors and an upper bound on the size of the graph. The computation proceeds in synchronized rounds in which each processor sends one unbounded message along each edge, which may be different for each edge. *Time* is measured by the number of rounds; local computation is free. At the end of the computation each node must report whether it is in the MIS, or which incident edge is part of the MM, or its assigned color. See [25, Ch. 1-2] for an extensive discussion of distributed models.

Prior Work: The vertex coloring, MM, and MIS problems have been the subject of intensive research since the mid-1980s [1], [2], [3], [9], [10], [11], [19], [20], [22], [23], [27], [30]. In 1986 Israeli and Itai [11] devised a randomized algorithm that computes an MM in $O(\log n)$ time with high probability,[2] and the same year Luby [22] and Alon, Babai, and Itai [1] independently proposed $O(\log n)$-time randomized MIS algorithms, which can also be used to compute $(\Delta+1)$-colorings in $O(\log n)$ time. These are the fastest known algorithms for MM and MIS on *general* graphs. It was recently shown that $(\Delta+1)$-coloring can be computed faster [30], in $O(\log\Delta+\sqrt{\log n})$ time w.h.p. Kuhn, Moscibroda, and Wattenhofer [18] proved that there exist n-vertex graphs with maximum degree $2^{\Theta(\sqrt{\log n})}$ on

** A full version of this paper is availabel online [6]. This work is supported by the US-Israel Binational Science Foundation grant No. 2008390, Israel Science Foundation grant No. 872009011, and NSF CAREER Grant No. CCF-0746673. Leonid Barenboim is supported by the Adams Fellowship Program of the Israel Academy of Sciences and Humanities. Additional funding was provided by Lynne and William Frankel Center for Computer Science.

[1] The MM problem is to compute a maximal set of vertex-disjoint edges. The MIS problem is to compute a maximal set of vertices, no two of which are adjacent. The $(\Delta+1)$-coloring problem is to assign colors from the palette $\{1,\ldots,\Delta+1\}$ such that no edge is monochromatic.

[2] *With high probability* (w.h.p.) means with probability $1-1/n^c$, for an arbitrarily large fixed constant c. All randomized algorithms cited in the paper finish their computation in the stated time bound, w.h.p.

0272-5428/12 $26.00 © 2012 IEEE
DOI 10.1109/FOCS.2012.60

IEEE computer society

which any algorithm for MM or MIS (even randomized) requires $\Omega(\sqrt{\log n})$ time. This implies a lower bound of $\Omega(\min\{\log\Delta, \sqrt{\log n}\})$ for these and many other problems. We will henceforth refer to this result as the *KMW bound*.

For deterministic algorithms the situation looks quite different. The fastest MIS and $(\Delta+1)$-coloring algorithms on general graphs run in $O(\Delta + \log^* n)$ time [3], [16] and $2^{O(\sqrt{\log n})}$ time [2], [27], whereas the fastest MM algorithms on general graphs run in $O(\Delta + \log^* n)$ time [26] and $O(\log^4 n)$ time [10]. For certain graph classes the bounds cited above can be improved. Barenboim and Elkin [3] showed that on graphs of arboricity λ, MM and MIS can be computed in time $O(\log n/\log\log n)$ for λ sufficiently small (λ must be less than $\log^{1-\epsilon} n$ for MM and less than $\log^{1/2-\epsilon} n$ for MIS.) We believe *arboricity* is an important graph parameter as it robustly captures the notion of *sparsity* without imposing any strict constraints. A graph has arboricity λ if its edge set can be covered by λ forests, or equivalently [24], if every subgraph has density less than λ.[3] Lenzen and Wattenhofer [19] gave a randomized MIS algorithm for unoriented trees ($\lambda = 1$) running in $O(\sqrt{\log n}\cdot\log\log n)$ time.[4] The MM and MIS problems on graphs of bounded growth have also been studied recently [9], culminating in an algorithm [30] running in $O(\log^* n)$ time on this graph class.

Faster coloring algorithms are known if one allows more than $(\Delta+1)$ colors. Linial [20] devised a deterministic $O(\Delta^2)$-coloring algorithm requiring $\log^* n + O(1)$ time, which was improved to $\frac{1}{2}\log^* n + O(1)$ by Szegedy and Vishwanathan [29]. Kothapalli et al. [15] gave a randomized $O(\Delta)$-coloring algorithm running in $O(\log\Delta + \sqrt{\log n})$ time, for all Δ, and Schneider and Wattenhofer [30] devised a randomized $O(\Delta+\log^{1+1/k} n)$-coloring algorithm running in time $O(k + \log^* n)$. Barenboim and Elkin [5] showed that $\Delta^{1+\epsilon}$-coloring can be computed in $O(\log\Delta\cdot\log n)$ time deterministically, for any $\epsilon > 0$. Graphs of bounded arboricity λ were shown [3], [5] to be amenable to faster coloring algorithms.

Our Results: We give a new randomized MM algorithm running in $O(\log\Delta + \log^4\log n)$ time, improving the 25-year old bound of $O(\log n)$ [11] and $O(\Delta + \log^* n)$ [26]. According to the KMW lower bound our algorithm is provably optimal whenever $\log\Delta \in [\log^4\log n, \sqrt{\log n}]$.

We give a randomized MIS algorithm running in $O(\log\Delta\cdot\sqrt{\log n})$ time, improving the 25-year old $O(\log n)$-time algorithms of Luby [22] and Alon, Babai, and Itai [1] when $\log\Delta \ll \sqrt{\log n}$. If $\Delta = (\log n)^{O(1)}$ we provide an even faster algorithm running in $2^{O(\sqrt{\log\log n})}$ time. These are the first general MIS algorithms running in sublogarithmic time for such a wide range of Δ.

For vertex coloring we give a $(\Delta+1)$-coloring algorithm running in $O(\log\Delta + 2^{O(\sqrt{\log\log n})})$ time, improving an $O(\log\Delta + \sqrt{\log n})$-time algorithm of Schneider and Wattenhofer [31]. As a result of this, we can now compute $O(\Delta)$-colorings in $2^{O(\sqrt{\log\log n})}$ time for all Δ, improving the $O(\sqrt{\log n})$-time algorithm of Kothapalli et al. [15].

As noted above, Δ is a significantly more sensitive graph parameter than the arboricity λ. We give a new technique for reducing the symmetry breaking problems on low arboricity graphs to low degree graphs, which is of independent interest. As direct corollaries, our reduction shows that MM and MIS can be solved in $O(\log\lambda + \sqrt{\log n})$ time and $O(\log\lambda\cdot\sqrt{\log n} + \log^{3/4} n)$ time, resp., on graphs with arboricity λ. In particular, for planar graphs (and, more generally, for graphs that exclude any fixed minor), our MM and MIS algorithms require only $O(\sqrt{\log n})$ and $O(\log^{2/3} n)$ time, respectively. (Naive substitution of $\lambda = O(1)$ into the $O(\log\lambda\sqrt{\log n} + \log^{3/4} n)$ bound for MIS yields $O(\log^{3/4})$. Improving that to $O(\log^{2/3} n)$ requires a different approach.) We also show that the KMW lower bound implies that MM in unoriented trees requires $\Omega(\sqrt{\log n})$ rounds. Hence our upper bound of $O(\sqrt{\log n})$ for MM in graphs with arboricity up to $2^{O(\sqrt{\log n})}$ is tight up to constant factors.

When λ is very small we give several algorithms that are faster for certain ranges of Δ. For example, when $\lambda = O(1)$ our MM, $(\Delta+1)$-coloring, and MIS algorithms run in time, respectively, $O(\log\Delta + \frac{\log\log n}{\log\log\log n})$, $O(\log\Delta + \log\log n)$, and $O(\log^2\Delta + \log\log n)$. These time bounds are exponentially faster, as a function of n, over previous deterministic algorithms [3], [5]. For the special case of trees ($\lambda = 1$) we give an even faster MIS algorithm whose running time is the minimum of $O(\sqrt{\log n\log\log n})$ and $O(\log\Delta\log\log\Delta + \frac{\log\log n}{\log\log\log n})$, which improves on [19].

See Figure 1 for a comparison of our results with prior work.

Technical Summary: All of our algorithms take the following two-phase approach. In Phase I we use some iterated randomized procedure that, with high probability, finds a large partial solution (a matching, independent set, or partial coloring) that effectively breaks the global problem into a collection of disjoint subproblems with $\mathrm{poly}(\log n)$ size or $O(\sqrt{\log n})$ diameter. In Phase II we solve each subproblem using the best available deterministic algorithm. It is for this reason that our running times are usually *exponentially* faster in terms of n than the best deterministic algorithms, e.g., a $2^{O(\sqrt{\log n})}$ bound becomes $2^{O(\sqrt{\log\log n})}$, a $\frac{\log n}{\log\log n}$ bound becomes $\frac{\log\log n}{\log\log\log n}$ and so on. This strategy has been used in other contexts, for example, in Beck's [7] algorithmic approach to the Lovász Local Lemma, the local hypergraph coloring algorithms of Rubinfeld et al. [28], and the $O(\Delta)$-coloring algorithm of Kothapalli et al. [15]. The main technical difficulty is in the analysis of Phase I's iterated randomized procedure.

[3]Note that many sparse graph classes have $\lambda = O(1)$, such as planar graphs, graphs avoiding a fixed minor, bounded genus graphs, and graphs of bounded degree or tree/pathwidth.

[4]The claimed time was $O(\sqrt{\log n\log\log n})$ but there was a flaw in the analysis. See Section 7 in [6].

Maximal Matching

Citation	Running Time	Graphs
II86 [11]	$\log n$	general
HKP01 [10]	$\log^4 n$ (Det.)	general
PR01 [26]	$\Delta + \log^* n$ (Det.)	general
BE08 [3]	$\frac{\log n}{\log\log n}$ (Det.)	$\lambda < \log^{1-\epsilon} n$
BE08 [3]	$\lambda + \log n$ (Det.)	general
L87 [20]	$\Omega(\log^* n)$ (Rand./Det.)	$\Delta \geq 2$
KMW04,10, [18]	$\Omega(\sqrt{\log n})$ (Rand./Det.) $\Omega(\log \Delta)$	general
This paper	$\log \Delta + \log^4 \log n$	general
	$\min \begin{cases} \log \lambda + \sqrt{\log n} \\ \log \Delta + \lambda + \log\log n \end{cases}$	all λ
	$\log \Delta + \frac{\log\log n}{\log\log\log n}$	$\lambda < \log^{1-\epsilon} \log n$

Maximal Independent Set

Citation	Running Time	Graphs
L,ABI [1], [22]	$\log n$	general
PS95 [27]	$2^{O(\sqrt{\log n})}$ (Det.)	general
BE09,K09 [4], [16]	$\Delta + \log^* n$ (Det.)	general
BE08 [3]	$\frac{\log n}{\log\log n}$ (Det.)	$\lambda < \log^{1/2-\epsilon} n$
BE09 [4]	$\lambda\sqrt{\log n}$ (Det.)	$\lambda > \sqrt{\log n}$
LW11 [19]	$\sqrt{\log n} \log\log n$	trees ($\lambda = 1$)
L87 [20]	$\Omega(\log^* n)$ (Rand./Det.)	$\Delta \geq 2$
KMW04,10, [18]	$\Omega(\sqrt{\log n})$ (Rand./Det.) $\Omega(\log \Delta)$	general
This paper	$\log \Delta \sqrt{\log n}$	general
	$2^{O(\sqrt{\log\log n})}$	$\Delta = \log^{O(1)} n$
	$\min \begin{cases} \log\lambda\sqrt{\log n} + \log^{3/4} n \\ \log^2 \Delta + \lambda \log \Delta + \lambda^\epsilon \log\log n \\ \log^2 \Delta + \lambda^{1+\epsilon} \log \Delta + \log\lambda \log\log n \end{cases}$	all λ
	$\log \Delta(\log \Delta + \frac{\log\log n}{\log\log\log n})$	$\lambda \leq \log^{1/2-\epsilon} \log n$
	$\log^{2/3} n$	$\lambda \leq \log^{1/3} n$
	$\min \begin{cases} \sqrt{\log n \log\log n} \\ \log \Delta \log\log \Delta + \frac{\log\log n}{\log\log\log n} \end{cases}$	trees ($\lambda = 1$)
	$\log \Delta \log\log n + 2^{O(\sqrt{\log\log n})}$	girth > 6

Vertex Coloring

Citation	Colors	Running Time	Notes
L87 [20]	$\Delta + 1$	$\Omega(\log^* n - \log^* \Delta)$	all $\Delta \geq 2$
BE09,K09 [4], [16]		$\Delta + \log^* n$ (Det.)	
PS95 [27]		$2^{O(\sqrt{\log n})}$ (Det.)	
SW10 [31]		$\log \Delta + \sqrt{\log n}$	
This paper		$\log \Delta + 2^{O(\sqrt{\log\log n})}$	
		$\log \Delta + \min \begin{cases} \lambda^{1+\epsilon} + \log\lambda \log\log n \\ \lambda + \lambda^\epsilon \log\log n \end{cases}$	all λ, fixed $\epsilon > 0$
This paper	$\Delta + O(\lambda)$	$\log \Delta + \lambda^\epsilon \log\log n$	all λ, fixed $\epsilon > 0$
This paper	$\Delta + \lambda^{1+\epsilon}$	$\log \Delta + \log\lambda \log\log n$	all λ, fixed $\epsilon > 0$
KSOS [15]	$O(\Delta)$	$\sqrt{\log n}$	
SW10 [31]		$k + \log^* n$	$k \leq \log\log n$, $\Delta > \log^{1+1/k} n$
BE10 [5]		$\Delta^\epsilon \log n$ (Det.)	fixed $\epsilon > 0$
This paper		$2^{O(\sqrt{\log\log n})}$	
BE10 [5]	$O(\lambda)$	$\lambda^\epsilon \log n$ (Det.)	fixed $\epsilon > 0$
BE10 [5]	$\Delta^{1+\epsilon}$	$\log \Delta \log n$ (Det.)	fixed $\epsilon > 0$
BE10 [5]	$\lambda^{1+\epsilon}$	$\log\lambda \log n$ (Det.)	fixed $\epsilon > 0$
SW10 [31]	$\Delta \log^{(k)} n$	k	$k \leq \log^* n, \Delta > \log^{1+1/k} n$
L92 [21]	Δ^2	$\log^* n$ (Det.)	

Figure 1. A summary of upper and lower bounds for MM, MIS, and vertex coloring. Here Δ is the maximum degree and λ the arboricity. All running times are randomized (w.h.p.) unless noted otherwise.

Our analyses often bound the running time in terms of Δ, which can be significantly larger than the arboricity λ. We give a new reduction that, roughly speaking, reduces the maximum degree to $\lambda \cdot 2^{\log^\epsilon n}$ in $O(\log^{1-\epsilon} n)$ time, for any $\epsilon \in (0, 1)$. This allows us to achieve sublogarithmic (in n) running times using algorithms that depend logarithmically on Δ.

Organization: Section II introduces some terminology and notation. Our MM, MIS, and $(\Delta + 1)$-coloring algorithms are presented in Sections III–V. Section VI presents a reduction from graphs with small arboricity to small degree.

II. Preliminaries

All logarithms are base 2 unless noted otherwise. The input graph is $G = (V, E)$. For any $V' \subseteq V$, let $G(V')$ be the subgraph of G induced by V'. Let $\Gamma_G(v) = \{u \,|\, (v, u) \in E\}$ and $\deg_G(v) = |\Gamma_G(v)|$ be the neighborhood of v in G and its cardinality. Let $\hat{\Gamma}_G(v) = \Gamma_G(v) \cup \{v\}$ be the neighborhood including v. Let $\Delta = \Delta(G) = \max_{v \in V} \deg_G(v)$

be the maximum degree. Let $\mathrm{dist}_G(u,v)$ be the length of the shortest path (i.e., distance) between u and v in G. The *diameter* of G is $\max_{u,v\in V}\mathrm{dist}_G(u,v)$ and the *weak diameter* of a subgraph $G(V')$ is the maximum distance between V'-vertices with respect to G, that is, $\max\{\mathrm{dist}_G(u,v) \mid u,v \in V'\}$. In a directed graph the indegree (outdegree) of v is the number of edges directed to (from) v, and the degree of v is the sum of its in and outdegree. A *forest* is an acyclic graph. An *oriented forest* is a directed forest in which each non-root has outdegree 1; a *pseudoforest* is a directed graph in which all vertices have outdegree 0 or 1.

In our analyses we use several standard concentration inequalities due to Chernoff, Janson, and Azuma-Hoeffding, given below. See [8] for their proofs.

Theorem 2.1: **(Chernoff with negative correlation)** Let $X = X_1+\cdots+X_n$ be the sum of n random variables, where the $\{X_i\}$ are independent or negatively correlated. Then for any $t > 0$:

$$\Pr[X \geq \mathrm{E}[X]+t], \Pr[X \leq \mathrm{E}[X]-t] \leq \exp\left(\frac{-2t^2}{\sum_i (a'_i - a_i)^2}\right)$$

where $a_i \leq X_i \leq a'_i$.

Theorem 2.2: **(Janson)** For $X = X_1 + \cdots + X_n$ the sum of n random variables and $t > 0$:

$$\Pr[X \geq \mathrm{E}[X]+t], \Pr[X \leq \mathrm{E}[X]-t] \leq \exp\left(\frac{-2t^2 \cdot (1/\chi)}{\sum_i (a'_i - a_i)^2}\right)$$

where $a_i \leq X_i \leq a'_i$ and χ is the fractional chromatic number of the dependency graph $G_X = (\{1,\ldots,n\}, \{(i,j) \mid X_i \text{ and } X_j \text{ are not independent}\})$.

Theorem 2.3: **(Azuma-Hoeffding)** A sequence $Y_0,\ldots,Y_n$ is a martingale with respect to $X_0,\ldots,X_n$ if Y_i is a function of $X_0,\ldots,X_i$ and $\mathrm{E}[Y_i \mid X_0,\ldots,X_{i-1}] = Y_{i-1}$. For such a martingale with bounded differences $a_i \leq Y_i - Y_{i-1} \leq a'_i$,

$$\Pr[Y_n > Y_0+t],\ \Pr[Y_n < Y_0-t] \leq \exp\left(-\frac{t^2}{2\sum_i (a'_i - a_i)^2}\right)$$

Corollary 2.4: Let $Z = Z_1 + \cdots + Z_n$ be the sum of n random variables and $X_0,\ldots,X_n$ be a sequence, where Z_i is a function of $X_0,\ldots,X_i$, $\mu_i = \mathrm{E}[Z_i \mid X_0,\ldots,X_{i-1}]$, $\mu = \sum_i \mu_i$, and $a_i \leq Z_i \leq a'_i$. Then

$$\Pr[Z > \mu + t],\ \Pr[Z < \mu + t] \leq \exp\left(-\frac{t^2}{2\sum_i (a'_i - a_i)^2}\right)$$

In our applications of these inequalities we often simplify the sum $\sum_i (a'_i - a_i)^2$ as follows. If $\sum_i (a'_i - a_i) \leq T$ and $\max_i (a'_i - a_i) \leq t$ then $\sum_i (a'_i - a_i)^2 \leq (T/t)t^2 = tT$.

III. An Algorithm for Maximal Matching

The Match procedure given below is a generalized version of the Israeli-Itai MM algorithm [11]. (See also [32].) It is given two vertex sets U_1, U_2 (not necessarily disjoint) and a matching M, and returns a matching on $U_1 \times U_2$ vertex-disjoint from M.

Match(U_1, U_2, M)

1) Initialize *directed* graphs $F_1 = (U_1 \cup U_2, \emptyset)$ and $F_2 = (U_1 \cup U_2, \emptyset)$.
2) Each $v \in U_1 \backslash V(M)$ chooses a neighbor $u \in U_2 \backslash V(M)$ uniformly at random and includes (v,u) in $E(F_1)$. (Note: F_1 is a pseudoforest.)
3) Each $u \in U_2$ with $\mathrm{indeg}_{F_1}(u) > 0$ chooses the $v' \in \{v : (v,u) \in E(F_1)\}$ with maximum node ID and includes (v',u) in $E(F_2)$. (Note: F_2 consists of directed paths and cycles.)
4) If $\deg_{F_2}(v) = 2$ then v chooses a bit $b(v) \in \{0,1\}$ uniformly at random. Otherwise $b(v) = 0$ (respectively, 1) if v is at the beginning (resp., end) of a path in F_2.
5) Return the matching $\{(v,u) \in E(F_2) : b(v) = 0 \text{ and } b(u) = 1\}$.

Note that U_1 and U_2 are allowed to contain matched vertices since these are specifically excluded in step 2. Phase I of our maximal matching algorithm consists of a sequences of $\Theta(\log\Delta)$ stages. In the pseudocode below M_i is the matching M just before stage i, $V_i = V\backslash V(M_i)$ is the set of unmatched vertices before stage i, and $\deg_i$ and Γ_i are the degree and neighborhood functions w.r.t. $G(V_i)$. Define the parameters δ_i, τ_i, and ν_i as

$$\delta_i = \frac{\Delta\sqrt{c_1 \ln n}}{\rho^i}, \quad \tau_i = \frac{2\Delta}{\rho^i\sqrt{c_1 \ln n}}, \quad \nu_i = \frac{\Delta^2}{\rho^{2i}} = \frac{\delta_i\tau_i}{2}$$

where c_1 is a sufficiently large constant and $\rho \approx 1.03$ a constant to be determined precisely later. Define $V_i^{\mathrm{lo}} = \{v \in V_i : \deg_i(v) \leq \tau_i\}$ and $V_i^{\mathrm{hi}} = \{v \in V_i : \deg_i(v) > \delta_i\}$ to be the low and high degree vertices at the beginning of stage i. In each stage i we supplement the current matching M_i first with a matching on $V_i^{\mathrm{lo}} \times V_i^{\mathrm{hi}}$, then with a matching on V_i.

Phase I: Initialize $M_0 \leftarrow \emptyset$ and execute stages $0,\ldots,c_2\log\Delta - 1$.
Stage i:
1. $M_{i+1} \leftarrow M_i \cup$ Match$(V_i^{\mathrm{lo}}, V_i^{\mathrm{hi}})$
2. $M_{i+1} \leftarrow M_{i+1} \cup$ Match(V_i, V_i)

Phase II: Let $\mathcal{C}$ be the connected components induced by $V_{c_2\log\Delta}$ with size at most $\log^9 n$. Deterministically compute a maximal matching $M(C)$ on each $C \in \mathcal{C}$ and return $M_{c_2\log\Delta} \cup \bigcup_{C\in\mathcal{C}} M(C)$.

The algorithm always returns a matching. If, at the beginning of Phase II, $\mathcal{C}$ contains all connected components on $V_{c_2\log\Delta}$ then the returned matching is clearly maximal. Thus, our goal is to show that with high probability, after Phase I there is no connected component of unmatched vertices with size at most $\log^9 n$. In the proof below $\deg(S)$ is short for $\sum_{u\in S}\deg(u)$ for $S \subset V$.

Lemma 3.1: Let l be any index for which $\tau_l > c_3 \ln n$ for a sufficiently large constant c_3. Then for all $i \in [0, l]$, $\deg_{i+1}(v) \le \delta_i$ and $\deg^{(2)}_{i+1}(v) \stackrel{\text{def}}{=} \deg_{i+1}(\Gamma_{i+1}(v)) \le \nu_i$ with probability $1 - 1/\text{poly}(n)$.

Proof: The two calls to Match in stage i are intended to maintain the two claimed properties: that v's degree degrades geometrically in each round and that the sum of v's neighbors' degrees degrades geometrically. The proof is by induction on i; the base case is trivial. For the sake of minimizing notation we use $\deg_i, \Gamma_i$, etc. to refer to the degree and neighborhood functions just before *each* call to Match in stage i. Consider a vertex $v \in V_i$ at the beginning of stage i. By the inductive hypothesis $\deg_i(v) \le \delta_{i-1}$ and $\deg^{(2)}_i(v) \le \nu_{i-1}$, from which it follows that v can have at most $\nu_{i-1}/\tau_i = \frac{\Delta\sqrt{c_1 \ln n}}{2\rho^{i-2}} = \delta_i \cdot (\rho^2/2)$ neighbors *not* in V^{lo}_i. If $\deg_i(v) > \delta_i$ (i.e., $v \in V^{\text{hi}}_i$) then in the first call to Match, v will be matched with probability[5] $1-(1-1/\tau_i)^{(1-\rho^2/2)\delta_i} > 1 - e^{(1-\rho^2/2)c_1 \ln n/2}$. By a union bound all vertices in V^{hi}_i are matched with probability at least $1 - 1/n^{c_1(1-\rho^2/2)/2-1} = 1/\text{poly}(n)$.

We now argue that after the second call to Match, $\deg^{(2)}_{i+1}(v) \le \nu_i$. Call a node *chosen* in the Match procedure if it has positive indegree in F_1. A node v will be guaranteed to have positive degree in F_2 if it is chosen *or* if it chooses an edge (v, u) and u has indegree 1 in F_1, i.e., u has *no choice* but to put (v, u) in F_2. Once in a path or cycle in F_2 the probability that v is matched is at least 1/2.

We evaluate the edges chosen by V_i-vertices for F_1 sequentially, beginning with all vertices outside of $\Gamma_i(v)$, then to each vertex in $\Gamma_i(v)$ one at a time, in *descending* order of node ID. (Recall that these were used for tie-breaking in Match.) Let $u \in \Gamma_i(v)$ be the current neighbor under consideration. If at least $\deg_i(u)/2$ neighbors of u are currently unchosen (by vertices already evaluated) then place u in set A, otherwise place u in set B. If u was put in set A and u *does* choose a previously unchosen neighbor (implying that it has positive degree in F_2) then also place u in set A'.

We first analyze the case that $\deg_i(A) \ge \deg^{(2)}_i(v)/2 \ge \nu_i/2$, then the case that $\deg_i(B) \ge \deg^{(2)}_i(v)/2$. (If $\deg^{(2)}_i(v) < \nu_i$ there is nothing to prove.) Observe that each vertex u, once in A, is moved to A' with probability at least 1/2, and if so, contributes $\deg_i(u) \le \delta_i$ to $\deg_i(A')$.[6] The probability that after evaluating each $u \in \Gamma_i(v)$, $\deg_i(A')$ is less than half its expectation is:

$$\begin{aligned}
&\Pr[\deg_i(A') < \tfrac{1}{2}\operatorname{E}[\deg_i(A')]] \\
&\le \Pr[\deg_i(A') < \tfrac{1}{4}\deg_i(A)] \\
&\le \exp(-\tfrac{(\frac{1}{4}\deg_i(A))^2}{2\sum_{u\in A}(\deg_i(u))^2}) && \{\text{Corollary 2.4}\} \\
&\le \exp(-\tfrac{1}{32}\tfrac{\deg_i(A)^2}{(\deg_i(A)/\delta_i)\delta_i^2}) && \{\deg_i(u) \le \delta_i\} \\
&\le \exp(-\tfrac{1}{32}\tfrac{\deg_i(A)}{\delta_i}) \\
&\le \exp(-\tfrac{1}{128}\tau_i) && \{\deg_i(A) \ge \nu_i/2 = \delta_i\tau_i/4\} \\
&= \exp(-\tfrac{\Delta}{64\rho^i\sqrt{c_1 \ln n}})
\end{aligned}$$

We proceed under the assumption that $\deg_i(A') \ge \frac{1}{4}\deg_i(A) \ge \nu_i/8$. Since each vertex with positive degree in F_2 is matched with probability at least 1/2, $\operatorname{E}[\deg_{i+1}(A')] \le \frac{1}{2}\deg_i(A')$. Moreover, whether $v \in A'$ is matched depends only on the bits selected by its neighbors in F_2, that is, the dependency graph of these events has chromatic number $\chi = 3$. Thus,

$$\begin{aligned}
&\Pr[\deg_{i+1}(A') > \tfrac{3}{4}\deg_i(A')] \\
&\le \exp(-\tfrac{2(\frac{1}{4}\deg_i(A'))^2}{\chi\cdot\sum_{u\in A'}(\deg_i(u))^2}) && \{\text{Theorem 2.2}\} \\
&\le \exp(-\tfrac{1}{24}\tfrac{\deg_i(A')^2}{(\deg_i(A')/\delta_i)\delta_i^2}) && \{\chi = 3, \deg_i(u) \le \delta_i\} \\
&\le \exp(-\tfrac{1}{24}\tfrac{\deg_i(A')}{\delta_i}) \\
&\le \exp(-\tfrac{1}{24}\tfrac{\tau_i}{16}) && \{\deg_i(A') \ge \nu_i/8 = \delta_i\tau_i/16\} \\
&\le \exp(-\tfrac{\Delta}{192\rho^i\sqrt{c_1 \ln n}})
\end{aligned}$$

We now turn to the case when $\deg_i(B) \ge \deg^{(2)}_i(v)/2 \ge \nu_i/2$. As each vertex $u \in B$ is evaluated at least $\deg_i(u)/2$ of its neighbors are already chosen. Let $C \subseteq \Gamma_i(B)$ be the set of chosen vertices in the second call to Match. For $x \in C$ let $d(x) \le \delta_i$ be the number of its neighbors in B and $d(C) = \sum_{x\in C} d(x)$. Thus, if x is matched then $\deg^{(2)}(v)$ is reduced by at least $d(x)$. It follows that $d(C) \ge \deg_i(B)/2 \ge \deg^{(2)}_i(v)/4 \ge \nu_i/4$ and therefore that $\operatorname{E}[\deg_{i+1}(B)] \le \deg_i(B) - d(C)/2 \le \frac{3}{4}\deg_i(B)$. We bound the probability that $\deg_{i+1}(B)$ deviates from its expectation using Janson's inequality, in exactly the same way as we handled $\deg_{i+1}(A')$. It follows that

$$\begin{aligned}
&\Pr[\deg_{i+1}(B) \ge \deg_i(B) - d(C)/4] \\
&\le \exp(-\tfrac{2(\frac{1}{4}d(C))^2}{\chi\cdot\sum_{x\in C} d(x)^2}) && \{\text{Theorem 2.2}\} \\
&\le \exp(-\tfrac{1}{24}\tfrac{d(C)^2}{(d(C)/\delta_i)\delta_i^2}) && \{\chi = 3, d(x) \le \delta_i\} \\
&\le \exp(-\tfrac{1}{24}\tfrac{\tau_i}{8}) && \{d(C) \ge \nu_i/4 = \delta_i\tau_i/8\} \\
&\le \exp(-\tfrac{\Delta}{96\rho^i\sqrt{c_1 \ln n}})
\end{aligned}$$

Regardless of whether $\deg_i(A) \ge \deg^{(2)}_i(v)/2$ or $\deg_i(B) \ge \deg^{(2)}_i(v)/2$, $\deg^{(2)}_{i+1}(v)$ is at most $\deg^{(2)}_i(v) -$

[5]Note that since $V^{\text{lo}}_i \cap V^{\text{hi}}_i = \emptyset$, F_1 consists of stars and F_2 consists of non-adjacent edges, all of which are added to the matching.

[6]Note that this process fits in the martingale framework of Corollary 2.4. Here X_j is the state of the system after evaluating the jth neighbor u of v and Z_j is $\deg_i(u)$ if u joins A' and 0 otherwise, which is a function of X_j. Thus, each Z_j has a range of at most δ_i.

$\deg_{i+1}(A')/4$ or $\deg_i^{(2)}(v) - d(C)/4$ with probability $1 - \exp(-\Omega(\tau_i)) = 1 - 1/\text{poly}(n)$. In either case $\deg_{i+1}^{(2)}(v) \leq (15/16)\deg_i^{(2)}(v)$. Setting $\rho = \sqrt{16/15}$ completes the induction. ∎

Lemma 3.1 implies that after $l = O(\ln(\Delta/\ln^{3/2} n))$ stages the maximum degree is at most $\delta_l = (c_1 \ln n/2)\tau_l = O(\ln^2 n)$. Lemma 3.2 implies that by the end of Phase I all surviving connected components have size $\text{poly}(\log n)$.

Lemma 3.2: At any point in Phase I, if the maximum degree in the graph induced by unmatched vertices is $\tilde{\Delta}$, then for some constant c_4, after $c_4 \log \tilde{\Delta}$ stages all connected components of unmatched vertices have size $\tilde{\Delta}^4 \log n$, with probability $1 - 1/\text{poly}(n)$.

Proof: The observations made in Lemma 3.1 imply that in each call to Match(V_i, V_i), each u loses a constant fraction of its neighbors (either because they are matched or u itself is matched) with constant probability. Moreover, the event that this occurs (a *success* for u) is independent of the success or failure of any u' at distance at least 5 from u.

We use the approach of [7], [28] to show that no components with size $> \tilde{\Delta}^4 \log n$ survive $c_4 \log \tilde{\Delta}$ stages. Consider a subgraph H of G with s vertices. One can easily see that there is some $V_0(H) \subseteq V(H)$ with $|V_0(H)| \geq s/(\tilde{\Delta}^4 + 1) = t$ such that for all $u, u' \in V_0(H)$, $\text{dist}(u, u') \geq 5$ and $\text{dist}(u, V_0(H)\backslash\{u\}) = 5$.[7] Such a set $V_0(H)$ corresponds to a tree with size t in the graph $G^5 = (V, \{(u, u') \mid \text{dist}(u, u') = 5\})$, which has maximum degree less than $\tilde{\Delta}^5$. There are fewer than 4^t distinct trees on t vertices and fewer than $n \cdot \tilde{\Delta}^{5t}$ ways to embed a tree on t vertices in G^5. For any one vertex the probability that it is not eliminated is at most the probability that it is not successful $O(\log \tilde{\Delta})$ times after $c_4 \log \tilde{\Delta}$ stages, which can be made $\tilde{\Delta}^{-c_5}$ for any c_5 by making c_4 sufficiently large. Since $V_0(H)$-vertices are at distance at least 5 from each other, these events are independent and the probability that H survives $c_4 \log \tilde{\Delta}$ stages is at most $\tilde{\Delta}^{-c_5 t}$. By a union bound, the probability that any such H survives is at most $n \cdot 4^t \cdot \tilde{\Delta}^{5t - c_5 t} = 1/\text{poly}(n)$ for $t = \log n$. ∎

The deterministic polylogarithmic-time algorithm of [10] and Lemma 3.2 imply the following result.

Theorem 3.3: A maximal matching can be computed in $O(\log \Delta + \log^4 \log n)$ time w.h.p. in an arbitrary distributed network.

IV. Maximal Independent Set Algorithms

To compute an MIS efficiently we employ the same general strategy used in our MM algorithm. We run a randomized algorithm (a variant of Luby's in this case) for a certain amount of time then argue that the connected components in the graph induced by vertices with degree at least $\Delta/2$ have weak diameter $O(\sqrt{\log n})$ (variant 1), or if $\Delta = \text{poly}(\log n)$, have size $O(\text{poly}(\log n))$ (variant 2). In the first case we use the trivial $O(\text{weak diameter}) = O(\sqrt{\log n})$ MIS algorithm and in the second we use the Panconesi-Srinivasan [27] algorithm, which runs in $2^{O(\sqrt{\log \log n})}$ time. Applying this halving algorithm $\log \Delta$ times reduces the maximum degree to zero. Since $O(\log \Delta \sqrt{\log n})$ is not an improvement over the logarithmic time MIS algorithms for $\Delta > 2^{\sqrt{\log n}}$, we assume in this section that $\Delta \leq 2^{\sqrt{\log n}}$.

Phase I of Halve computes independent sets $I_0 = \emptyset \subseteq I_1 \subseteq \cdots \subseteq I_\kappa$ and Phase II computes an MIS on the components of high-degree ($\geq \Delta/2$) vertices in $V\backslash\hat{\Gamma}(I_\kappa)$. In stage i the *active* vertices are $A_i = V\backslash\hat{\Gamma}(I_i)$ and $\deg_i$ and Γ_i are the functions with respect to A_i.

Halve—Phase I:
Initialize $I_0 \leftarrow \emptyset$ and execute stages $0, 1, \ldots, \kappa - 1$ where $\kappa = c_6\sqrt{\log n}$ (variant 1) or $c_6 \log \Delta$ (variant 2).
Stage i:
1. Each $v \in A_i$ *selects* itself with probability $\frac{1}{\Delta+1}$.
2. $I_{i+1} \leftarrow I_i \cup \{v \mid v$ is the only vertex in $\hat{\Gamma}_i(v)$ that selects itself$\}$.

Halve—Phase II:
Let $U = \{v \in A_\kappa : \deg_\kappa(v) \geq \Delta/2\}$ be the set of active vertices with degree at least $\Delta/2$. Let $\mathcal{C}$ be the set of connected components of $G(U)$ with weak diameter less than $5\sqrt{\log n}$ (variant 1) or size less than $\Delta^4 \log n$ (variant 2). Deterministically compute an MIS $I(C)$ for each $C \in \mathcal{C}$ and return $I_\kappa \cup \bigcup_{C \in \mathcal{C}} I(C)$.

The proof of the following lemma is omitted due to space constraints. See the full version of this paper [6].

Lemma 4.1: Let $S \subseteq A_i$ be such that $\text{dist}(u, u') \geq 5$ and $\deg_i(u) \geq \Delta/2$ for all $u, u' \in S$. The probability that $S \subseteq A_{i+1}$ is less than $p^{|S|}$ where $p = 1 - (1 - e^{-1/2})e^{-1} \approx 0.85$.

In order to prove that U induces components with weak diameter less than $5\sqrt{\log n}$ it suffices to prove that for each $u, u' \in V$ at distance at least $5(\sqrt{\log n} - 1)$, every path from u to u' contains some vertex not in U. To that end we define $\mathcal{P}$ to be the set of *all* paths (not necessarily shortest) between pairs of vertices at distance at least $5(\sqrt{\log n} - 1)$. We first claim that each $P = (u_1, \ldots, u_r) \in \mathcal{P}$ contains a $Q(P) = \{q_1, \ldots, q_{\sqrt{\log n}}\} \subset V(P)$ such that $\text{dist}(u, u') \geq 5$ for all $u, u' \in Q(P)$. We generate $Q(P)$ one vertex at a time maintaining the invariant that $\text{dist}(q_j, u_r) \geq 5(\sqrt{\log n} - j)$. Define $q_1 = u_1$ and once q_j is known, define $q_{j+1} = u_k$ where k is the maximum index such that $\text{dist}(q_j, u_k) = 5$. It follows that for all $j' < j$, $\text{dist}(q_{j'}, q_{j+1}) > 5$. By the triangle inequality, $\text{dist}(q_j, u_r) \geq 5(\sqrt{\log n} - j)$ implies $\text{dist}(q_{j+1}, u_r) \geq 5(\sqrt{\log n} - (j + 1))$.

Define $\mathcal{Q} = \{Q(P) \mid P \in \mathcal{P}\}$ and $\mathcal{W}$ to be the set of all walks of length exactly $5(\sqrt{\log n} - 1)$. Every $Q(P) \in \mathcal{Q}$ can be mapped injectively to a walk in $\mathcal{W}$ by taking the concatenation of arbitrary shortest paths between successive

[7] For example, repeatedly select a vertex u in $V(H)$ at distance 5 from some previously selected vertex, then remove all vertices within distance 4 of u.

vertices in $Q(P)$. Thus, $|\mathcal{Q}| \leq |\mathcal{W}| \leq n\Delta^{5(\sqrt{\log n}-1)}$.

Lemma 4.2: After $\kappa = c_6\sqrt{\log n}$ (or $\kappa = c_6 \log \Delta$) stages of Phase I, U induces components with weak diameter less than $5\sqrt{\log n}$ (or size less than $\Delta^4 \log n$) w.h.p., for a sufficiently large c_6.

Proof: If the weak diameter criterion is violated then there is some $P \in \mathcal{P}$ with $V(P) \subseteq U$. By Lemma 4.1, in each stage i that all $Q(P)$-vertices have degree at least $\Delta/2$ the probability that none become inactive in stage $i+1$ is $\exp(-\Omega(|Q(P)|)) = \exp(-\Omega(\sqrt{\log n}))$. Thus, after κ iterations the probability that $Q(P) \subseteq U$ is $\epsilon = \exp(-\Omega(\sqrt{\log n}) \cdot \kappa) = \exp(-\Omega(c_6 \log n))$. By a union bound, the probability that any $Q \in \mathcal{P}$ has $Q(P) \subseteq U$ is $\epsilon|\mathcal{Q}| < \epsilon n\Delta^{5(\sqrt{\log n}-1)} < \epsilon n^6$ (since $\Delta \leq 2^{\sqrt{\log n}}$), which is $1/\mathrm{poly}(n)$ for sufficiently large c_6. It follows that no $P \in \mathcal{P}$ has $V(P) \subseteq U$ with probability $1 - 1/\mathrm{poly}(n)$ and that Phase II successfully makes inactive all vertices in U.

The analysis of the second variant of the algorithm follows that of Lemma 3.2. Each connected subgraph with s vertices contains a vertex set with size $t \geq s/(\Delta^4+1)$ that forms a tree in $G^5 = (V, \{(u,u') \mid \mathrm{dist}(u,u') = 5\})$ with size t. There are at most $n4^t\Delta^{5(t-1)}$ trees embedded in G^5 with size t. If each of the t vertices has degree $\Delta/2$ in stage i, by Lemma 4.1 the probability that all t are active in stage $i+1$ is $\exp(-\Omega(t))$ and the probability that all are in U is $\epsilon = \exp(-\Omega(t\kappa)) = \exp(-\Omega(c_6 t \log \Delta))$. By a union bound the probability that a component with size $s = t\Delta^4$ exists in U is less than $\epsilon n 4^t \Delta^{5(t-1)}$, which is $1/\mathrm{poly}(n)$ for $t = \log n$ and sufficiently large c_6. ∎

Our main result of this section now follows from Lemma 4.2 and the algorithm of Panconesi and Srinivasan [27].

Theorem 4.3: An MIS can be computed in $O(\log \Delta \sqrt{\log n})$ time w.h.p. in an arbitrary distributed network, or in $\exp(O(\sqrt{\log \log n}))$ time w.h.p. when $\Delta = \mathrm{poly}(\log n)$.

In the full version of this paper [6] we also prove the following bounds for MIS in graphs of large girth. These results generalize and slightly improve results of Lenzen and Wattenhofer [19]. Our proof of these results is based to a large extent on the proof of [19].

Theorem 4.4: On graphs of girth greater than 6, an MIS can be computed in time on the order of

$$\log \Delta \cdot \log \log n + \exp\{O(\sqrt{\log \log n})\}$$

Moreover, an MIS of an unoriented tree can be computed in time

$$\min\left\{\sqrt{\log n \log \log n},\ \log \Delta \cdot \log \log \Delta + \frac{\log \log n}{\log \log \log n}\right\}$$

V. A $(\Delta+1)$-Coloring Algorithm

Schneider and Wattenhofer [31] presented a randomized $(\Delta+1)$-coloring algorithm running in $O(\log \Delta + \sqrt{\log n})$ time and several faster $O(\Delta)$-coloring algorithms assuming $\Delta = \Omega(\log n)$. Here we give a faster $(\Delta+1)$-coloring algorithm running in $O(\log \Delta + \exp(O(\sqrt{\log \log n})))$ time, which also implies that a graph can be $O(\Delta)$-colored in $\exp(O(\sqrt{\log \log n}))$ time, for any Δ.[8]

Theorem 5.1: A $(\Delta+1)$-coloring can be computed in $O(\log \Delta + \exp(O(\sqrt{\log \log n})))$ time w.h.p. in an arbitrary distributed network.

Due to space constraints we can only provide a sketch of the algorithm and analysis; see [6] for a complete description. Phase I of the algorithm takes the most natural randomized approach [13]. Let $\Psi = \{1, \ldots, \Delta+1\}$ be the palette. Let $c_i : V \to \Psi \cup \{\perp\}$ be the partial coloring before the ith stage of Phase I, where $\perp$ indicates no color, and let $\Gamma_i(v) = \{u \in \Gamma(v) \mid c_i(u) = \perp\}$ be the uncolored neighborhood of v. In the ith stage each colored vertex retains its color and each uncolored vertex v selects a color $c'(v)$ uniformly at random from its available palette $\Psi_i(v) \stackrel{\mathrm{def}}{=} \Psi \backslash \{c(u) \mid u \in \Gamma(v)\}$. It sets $c_{i+1}(v) = c'(v)$ if $c'(v) \notin \{c'(u) \mid u \in \Gamma_i(v)\}$.

We first prove that in each stage of Phase I, each vertex is colored with constant probability. This does not imply that a constant fraction of a vertex v's neighborhood $\Gamma_i(v)$ is colored with probability $\exp(-\Omega(\deg_i(v)))$ as the relevant events are not independent. They are, however, negatively correlated, which allows us to invoke Theorem 2.1. In particular, we prove that in each stage, each *high degree* vertex (having degree $\Omega(\log n)$) loses a constant fraction of its high degree neighbors. Therefore, after $O(\log \Delta)$ stages all vertices have at most $O(\log n)$ neighbors with degree $\Omega(\log n)$, though there is no upper bound on the maximum degree. The subgraph induced by non-high degree vertices has, by definition, maximum degree $O(\log n)$. Once this subgraph is colored the remaining subgraph of uncolored vertices *also* has maximum degree $O(\log n)$. Thus, in Phase II we must solve two subproblems (sequentially) on graphs with maximum degree $\tilde{\Delta} = O(\log n)$. Consider one such subproblem. The argument employed in Theorems 3.3 and 4.3 shows that after $O(\log \tilde{\Delta}) = O(\log \log n)$ further stages of the randomized coloring algorithm, all components of uncolored vertices have size $s = \tilde{\Delta}^4 \log n = O(\log^5 n)$. These components can be colored deterministically in $\exp(O(\sqrt{\log s})) = \exp(O(\sqrt{\log \log n}))$ time using the algorithm of [27].

VI. Bounded Arboricity Graphs

Recall that a graph has arboricity λ if its edge set is the union of λ forests. In the proofs of Lemma 6.1 and Theorem 6.2, $\deg_{E'}(u)$ is the number of edges incident to u in $E' \subseteq E$ and $\deg_{V'}(u) = \deg_{G(V' \cup \{u\})}(u)$ is the number of neighbors of u in $V' \subseteq V$. Due to space constraints the proof of the following technical lemma is omitted. (See [6].)

Lemma 6.1: Let G be a graph of m edges, n vertices, and arboricity λ.

[8] If $\Delta > \log n$ use Schneider-Wattenhofer [31]; if $\Delta < \log n$ use our algorithm.

1) $m \le \lambda n$.
2) The number of vertices with degree at least $t \ge \lambda + 1$ is at most $\lambda n/(t-\lambda)$.
3) The number of edges whose endpoints both have degree at least $t \ge \lambda + 1$ is at most $\lambda m/(t-\lambda)$.

Theorem 6.2: Let G be a graph of arboricity λ and $t \ge \max\{11^8 \cdot \lambda^8, (4(c+1)\ln n)^7\}$ be a parameter. In $O(\log_t n)$ time we can find a matching $M \subseteq E(G)$ (or an independent set $I \subseteq V(G)$) such that with probability at least $1 - 1/n^c$, the maximum degree in the induced graph $G(V \backslash V(M))$ (or $G(V \backslash \hat{\Gamma}(I))$) is at most $t\lambda$.

Proof: In $O(\log_t n)$ rounds we commit edges to M (or vertices to I) and remove all incident edges (or incident vertices). Let G be the graph still under consideration before some round and let $\mathcal{H} = \{v \in V \mid \deg_G(v) \ge t\lambda\}$ be the remaining high-degree vertices. Our goal is to reduce the size of $\mathcal{H}$ by a roughly $t^{1/7}$ factor. Let $\mathcal{J} = \{v \in \mathcal{H} \mid \deg_{\mathcal{H}}(v) \ge t\lambda/2\}$. It follows that any vertex $v \in \mathcal{H}' = \mathcal{H} \backslash \mathcal{J}$ has $\deg_{V \backslash \mathcal{H}}(v) \ge t\lambda/2$. Let $\tilde{E} \subset E(\mathcal{H}', V \backslash \mathcal{H})$ be any set of edges such that for $v \in \mathcal{H}'$, $\deg_{\tilde{E}}(v) = t\lambda/2$ (that is, discard all but $t\lambda/2$ edges arbitrarily) and let $\mathcal{S} = \{u \mid v \in \mathcal{H}' \text{ and } (v,u) \in \tilde{E}\}$ be the neighborhood of $\mathcal{H}'$ with respect to $\tilde{E}$. Note that $|\mathcal{S}| \le t\lambda|\mathcal{H}'|/2$. See Figure 2. We define bad $\mathcal{S}$-vertices, bad

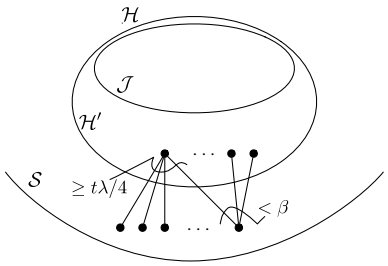

Figure 2. Good $\mathcal{S}$-vertices have fewer than β neighbors in $\mathcal{H}'$ and fewer than β^2 neighbors in $\mathcal{S}$. Good $\mathcal{H}'$-vertices have at least $t\lambda/4$ good neighbors in $\mathcal{S}$.

$\tilde{E}$-edges, and bad $\mathcal{H}'$-vertices as follows, where $\beta = t^{1/7}$. Let $B_{\mathcal{S}} = \{u \in \mathcal{S} \mid \deg_{\tilde{E}}(u) \ge \beta \text{ or } \deg_{\mathcal{S}}(u) \ge \beta^2\}$, let $B_{\tilde{E}} = \{(v,u) \in \tilde{E} \mid u \in B_{\mathcal{S}}\}$, and let $B_{\mathcal{H}'} = \{v \in \mathcal{H}' \mid \deg_{\tilde{E} \backslash B_{\tilde{E}}}(v) < \lambda t/4\}$.

By Lemma 6.1(3) the number of bad $(v,u) \in B_{\tilde{E}}$ due to $\deg_{\tilde{E}}(u) \ge \beta$ is at most $\lambda|\tilde{E}|/(\beta - \lambda) \le \lambda(t\lambda|\mathcal{H}'|/2)/(\beta - \lambda)$. By Lemma 6.1(2) the number of *additional* bad $(v,u) \in B_{\tilde{E}}$ due to $\deg_{\mathcal{S}}(u) \ge \beta^2$ is at most $(\beta-1)\lambda|\mathcal{S}|/(\beta^2 - \lambda) \le (\beta-1)\lambda(t\lambda|\mathcal{H}'|/2)/(\beta^2 - \lambda)$ since there are at most $\lambda|\mathcal{S}|/(\beta^2 - \lambda)$ such u and each contributes fewer than β edges in $\tilde{E}$. In total $|B_{\tilde{E}}| < 1.1 \cdot \lambda^2 t|\mathcal{H}'|/\beta$. (Here we use that $\beta = t^{1/7} > 11 \cdot \lambda$, by an assumption of the theorem.) Note that a bad $v \in \mathcal{H}'$ must be incident to more than $t\lambda/4$ edges in $B_{\tilde{E}}$ since $\deg_{\tilde{E}}(v) = t\lambda/2$. Hence $|B_{\mathcal{H}'}| < |B_{\tilde{E}}|/(t\lambda/4) < 4.4 \cdot \lambda|\mathcal{H}'|/\beta = 4.4 \cdot \lambda|\mathcal{H}'|/t^{1/7}$.

Our goal now is to select some vertices for the MIS (or edges for the MM) that eliminate all good $\mathcal{H}'$ vertices, with high probability. Each vertex $u \in \mathcal{S} \backslash B_{\mathcal{S}}$ selects a random number and joins the MIS if it holds a local maximum. The probability that u joins the MIS is $1/(\deg_{\mathcal{S} \backslash B_{\mathcal{S}}}(u) + 1) \ge 1/\beta^2$ and this event is clearly independent of all $u' \in \mathcal{S} \backslash B_{\mathcal{S}}$ at distance (in $\mathcal{S} \backslash B_{\mathcal{S}}$) at least 3. Note that since the maximum degree in the graph induced by $\mathcal{S} \backslash B_{\mathcal{S}}$ is less than β^2, the neighborhood of any good $v \in \mathcal{H}' \backslash B_{\mathcal{H}'}$ contains a subset of at least $(t\lambda/4)/\beta^4$ vertices, each pair of which is at distance at least 3 with respect to $\mathcal{S} \backslash B_{\mathcal{S}}$. (Such a set could be selected greedily.) Thus, the probability that no neighbor of $v \in \mathcal{H}' \backslash B_{\mathcal{H}'}$ joins the MIS is at most $(1 - 1/\beta^2)^{t\lambda/(4\beta^4)} < e^{-t\lambda/(4\beta^6)} = e^{-t^{1/7}\lambda/4} < 1/n^{c+1}$. Thus, with high probability all good $\mathcal{H}' \backslash B_{\mathcal{H}'}$ vertices are eliminated. Any remaining high degree vertices must be in $\mathcal{J}$ or $B_{\mathcal{H}'}$, which, by Lemma 6.1(2) number at most $\lambda|\mathcal{H}|/(t/2 - \lambda) + 4\lambda|\mathcal{H}'|/t^{1/7} < 5\lambda|\mathcal{H}|/t^{1/7}$. The number of high-degree vertices is reduced by a $t^{\Omega(1)}$ factor since $t > \max\{11^8 \cdot \lambda^8, (4(c+1)\ln n)^7\}$, so after $O(\log_t n)$ rounds all high-degree vertices have been eliminated, with probability at least $1 - 1/n^c$.

The case of MM can be argued in a similar way. ∎

Theorem 6.3: Given a graph of arboricity λ, an MM can be computed in time on the order of:

$$\min\left\{ \log\lambda + \sqrt{\log n}, \quad \log\Delta + \lambda + \log\log n \right\}$$

for all λ, and in $O\left(\log\Delta + \frac{\log\log n}{\log\log\log n}\right)$ time for $\lambda < \log^{1-\Omega(1)}\log n$.

Proof: The second and third bounds follow from Theorem 3.3 by substituting for [10] the deterministic MM algorithms of Barenboim and Elkin [3] for small arboricity graphs. Their algorithms run in $O(\frac{\log s}{\log\log s})$ time on graphs with size s and arboricity $\lambda < \log^{1-\Omega(1)} s$ and in time $O(\lambda + \log s)$ in general. In the context of our algorithm, $s \le \log^9 n$. The first MM bound is a consequence of Theorem 6.2 and Theorem 3.3. We reduce the maximum degree to $\Delta = \lambda t = \lambda \cdot \max\{11^8 \cdot \lambda^8, 2^{\sqrt{\log n}}\}$ in $O(\log_t n) = O(\sqrt{\log n})$ time and find an MM of the resulting graph in $O(\log\Delta + \log^4\log n) = O(\log\lambda + \sqrt{\log n})$ time. ∎

Note that, in particular, for graphs of constant arboricity (e.g., planar graphs or graphs that exclude a fixed minor), the algorithm in Theorem 6.3 constructs an MM within $O(\sqrt{\log n})$ time. The following theorem is proved similarly to Theorem 6.3, except that t is set to $2^{\log^{1/4} n}$. See [6] for full proof.

Theorem 6.4: Given a graph of arboricity λ, an MIS can be computed in time on the order of:

$$\min\left\{ \begin{array}{l} \log\lambda\sqrt{\log n} + \log^{3/4} n, \\ \log^2\Delta + \lambda\log\Delta + \lambda^{\epsilon}\log\log n, \\ \log^2\Delta + \lambda^{1+\epsilon}\log\Delta + \log\lambda\log\log n \end{array} \right\}$$

for all λ and $\epsilon > 0$, and in $O\left(\log\Delta\left(\log\Delta + \frac{\log\log n}{\log\log\log n}\right)\right)$ time for $\lambda < \log^{1/2-\Omega(1)} \log n$.

In particular, the first bound of Theorem 6.4 implies that for graphs of small and moderate arboricity ($\lambda \leq 2^{\log^{1/4} n}$), an MIS can be computed in $O(\log^{3/4} n)$ time. Moreover, next we argue that when the arboricity is small (specifically, $\lambda \leq \log^{1/3} n$) then an MIS can be computed even faster than that.

Run the degree reduction algorithm from Theorem 6.2 with $t = 2^{\log^{1/3} n}$. As a result we reduce the problem to an MIS in graphs with maximum degree $\Delta' = \lambda \cdot 2^{\log^{1/3} n}$, within $O(\log^{2/3})$ time. Now we invoke the MIS algorithm given by the second bound of Theorem 6.4. Its running time is $\log^2 \Delta' + \lambda \cdot \log \Delta' + \lambda^\epsilon \cdot \log\log n \leq (\log\lambda + \log^{1/3} n)^2 + \lambda(\log\lambda + \log^{1/3} n) + \lambda^\epsilon \cdot \log\log n = O(\log^{2/3} n)$.

Theorem 6.5: In a graph of arboricity $\lambda = O(\log^{1/3} n)$, an MIS can be computed in $O(\log^{2/3} n)$ time.

Due to space constraints we skip the (simple) proof of the next theorem.

Theorem 6.6: Given a graph of arboricity λ and any fixed $\epsilon > 0$, a $(\Delta + \lambda^{1+\epsilon})$-coloring can be computed in $O(\log\Delta + \log\lambda\log\log n)$ time and a $(\Delta + O(\lambda))$-coloring can be computed in $O(\log\Delta + \lambda^\epsilon \log\log n)$ time. Consequently, a $(\Delta+1)$-coloring can be computed in time

$$O(\log\Delta + \min\{\lambda^{1+\epsilon} + \log\lambda\log\log n,\ \ \lambda + \lambda^\epsilon \log\log n\}).$$

In particular, in graphs of constant arboricity, $(\Delta+1)$-coloring can be computed in just $O(\log\Delta + \log\log n)$ time.

Our MM algorithm from Theorem 6.3 runs in $O(\sqrt{\log n})$ time for every arboricity λ in the range $1 \leq \lambda = 2^{O(\sqrt{\log n})}$. We argue that this bound is optimal even for *constant* λ by appealing to the KMW lower bound of [17], [18]. In [18] it is shown that there exist constant $0 < c', c$ such that any (possibly randomized) algorithm for computing approximate minimum vertex cover (henceforth, MVC) in graphs with girth[9] at least $c' \cdot \sqrt{\log n}$ which runs for $c \cdot \sqrt{\log n}$ rounds or less ($3c < c'$) has a super-constant expected approximation ratio. By way of a standard reduction from 2-approximate MVC to MM, which we review below, they observe that computing MM also requires $\Omega(\sqrt{\log n})$ rounds in expectation. Our goal is showing a similar bound for graphs of constant arboricity, which does not follow directly from [18] as their hard graphs have arboricity $2^{O(\sqrt{\log n})}$. As a first step, we show that any MM algorithm on *general graphs* that succeeds *with high probability* requires $\Omega(\sqrt{\log n})$ time.

Suppose, for the purpose of obtaining a contradiction, that there exists an MM algorithm running in time $c\sqrt{\log n}$ on the KWM graph that fails with probability at most $p(n) = 1/n$. To obtain an approximate MVC algorithm, run the MM algorithm for $c\sqrt{\log n}$ rounds. Any matched vertex joins the vertex cover as well as any vertex that detects a local violation, namely a vertex incident to two matched edges or an unmatched vertex incident to another unmatched vertex. As the minimum vertex cover is at least the size of any maximal matching, the expected approximation ratio of this algorithm is at most $2 \cdot \Pr[\text{no failure occurs}] + n \cdot \Pr[\text{some failure occurs}] \leq 2 + n \cdot \frac{1}{n} = 3$, a contradiction. Hence there is no algorithm that runs for $c \cdot \sqrt{\log n}$ rounds in graphs with girth at least $c' \cdot \sqrt{\log n}$, $3c < c'$, that computes an MM with probability at least $1 - 1/n$.

We use an indistinguishability argument to show that the $\Omega(\sqrt{\log n})$ lower bound also holds for MM on graphs with constant arboricity, even trees. Observe that to show a lower bound for a randomized algorithm, it is enough to prove the same lower bound under the assumption that the identities of graph vertices were selected independently and uniformly at random, from, say, $[1, n^{10}]$. (These new identity numbers can be tossed before the computation starts.) Suppose there is, in fact, an algorithm that given a tree with a random (in the above sense) assignment of identities, constructs an MM within $c \cdot \sqrt{\log n}$ rounds with success probability at least $1 - 1/n^2$. Run this algorithm for $c \cdot \sqrt{\log n}$ rounds on the KMW graph G with girth $c' \cdot \sqrt{\log n}$, assuming random assignment of identities in G. Due to the girth bound, the view of every vertex in G is identical to its view in a tree, and thus from its perspective a correct MM will be computed with probability at least $1 - 1/n^2$. By a union bound, a correct MM for the graph G will be computed with probability at least $1 - 1/n$, a contradiction.

Corollary 6.7: Any MM algorithm for n-vertex unoriented trees that runs for at most $c \cdot \sqrt{\log n}$ rounds, for some universal constant $c > 0$, has failure probability at least $1/n^2$.

The failure probability in Corollary 6.7 can be made arbitrarily close to 1 by considering a graph consisting of the union of $n^{1-\epsilon}$ such trees, each with size n^ϵ, for some $\epsilon > 0$. Note that Corollary 6.7 does not extend to the MIS problem on trees, even though MIS appears to be just as difficult as MM. The $\Omega(\sqrt{\log n})$ lower bound for MIS from [17], [18] is obtained by considering the line graph of a KWM graph, which has girth 3 rather than $\Theta(\sqrt{\log n})$.

[9]The girth is the length of the shortest cycle.

REFERENCES

[1] N. Alon, L. Babai, and A. Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *Journal of Algorithms*, 7(4):567–583, 1986.

[2] B. Awerbuch, A. V. Goldberg, M. Luby, and S. Plotkin. Network decomposition and locality in distributed computation. In *Proc. of the 30th Symposium on Foundations of Computer Science*, pages 364–369, 1989.

[3] L. Barenboim, and M. Elkin. Sublogarithmic distributed MIS algorithm for sparse graphs using Nash-Williams decomposition. In *Proc. of the 27th ACM Symp. on Principles of Distributed Computing*, pages 25–34, 2008.

[4] L. Barenboim, and M. Elkin. Distributed $(\Delta+1)$-coloring in linear (in Δ) time. In *Proc. of the 41st ACM Symp. on Theory of Computing*, pages 111-120, 2009. Full version available at *http://arXiv.org/abs/0812.1379v2*.

[5] L. Barenboim, and M. Elkin. Deterministic distributed vertex coloring in polylogarithmic time. In *Proc. of the 29th ACM Symp. on Principles of Distributed Computing*, pages 410-419, 2010.

[6] L. Barenboim, M. Elkin, S. Pettie, and J. Schnedier. The Locality of Distributed Symmetry Breaking. *http://arXiv.org/abs/1202.1983v2*, 2012.

[7] J. Beck. An Algorithmic Approach to the Lovász Local Lemma. I. *Random Struct. Algorithms* 2(4):343–366, 1991.

[8] D. P. Dubhashi and A. Panconesi. Concentration of Measure for the Analysis of Randomized Algorithms. Cambridge University Press, 2009.

[9] B. Gfeller, and E. Vicari. A randomized distributed algorithm for the maximal independent set problem in growth-bounded graphs. In *Proc. of the 26th ACM Symp. on Principles of Distributed Computing*, pages 53-60, 2007.

[10] M. Hanckowiak, M. Karonski, and A. Panconesi. On the Distributed Complexity of Computing Maximal Matchings. *SIAM J. Discrete Math.* 15(1):41–57, 2001.

[11] A. Israeli, and A. Itai. A fast and simple randomized parallel algorithm for maximal matching. *Information Processing Letters* 22(2):77-80, 1986.

[12] S. Janson. Large deviations for sums of partly dependent random variables. *Random Struct. Algorithms* 24(3):234–248, 2004.

[13] O. Johansson. Simple Distributed $\Delta+1$-coloring of Graphs. *Inf. Process. Lett.* 70(5):229–232, 1999.

[14] K. Kothapalli, and S. Pemmaraju. Distributed graph coloring in a few rounds. In *Proc. of the 30th ACM Symp. on Principles of Distributed Computing*, pages 31-40, 2011.

[15] K. Kothapalli, C. Scheideler, M. Onus, and C. Schindelhauer. Distributed coloring in $O(\sqrt{\log n})$ bit rounds. In *Proc. of the 20th International Parallel and Distributed Processing Symposium*, 2006.

[16] F. Kuhn. Weak graph colorings: distributed algorithms and applications. In *Proc. of the 21st ACM Symposium on Parallel Algorithms and Architectures*, pages 138–144, 2009.

[17] F. Kuhn, T. Moscibroda, and R. Wattenhofer. What cannot be computed locally! In *Proc. of the 23rd ACM Symp. on Principles of Distributed Computing*, pages 300-309, 2004.

[18] F. Kuhn, T. Moscibroda, and R. Wattenhofer. Local Computation: Lower and Upper Bounds. *http://arXiv.org/abs/1011.5470*, 2010.

[19] C. Lenzen and R. Wattenhofer. MIS on trees. In *Proc. of the 30th ACM Symp. on Principles of Distributed Computing*, pages 41-48, 2011.

[20] N. Linial. Distributive graph algorithms: Global solutions from local data In *Proc. of the 28th Annual Symp. on Foundation of Computer Science*, pages 331–335, 1987.

[21] N. Linial. Locality in distributed graph algorithms. *SIAM Journal on Computing*, 21(1):193–201, 1992.

[22] M. Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM Journal on Computing*, 15:1036-1053, 1986.

[23] Y. Métivier, J. M. Robson, N. Saheb-Djahromi, and A. Zemmari. An optimal bit complexity randomized distributed MIS algorithm *Distributed Computing*, 23(5-6):331–340, 2011.

[24] C. Nash-Williams. Decompositions of finite graphs into forests. *J. London Math*, 39:12, 1964.

[25] D. Peleg. *Distributed Computing: A Locality-Sensitive Approach*. SIAM, 2000.

[26] A. Panconesi, and R. Rizzi. Some simple distributed algorithms for sparse networks. *Distributed computing*, 14(2):97–100, 2001.

[27] A. Panconesi, and A. Srinivasan. On the complexity of distributed network decomposition. *Journal of Algorithms*, 20(2):581-592, 1995.

[28] R. Rubinfeld, G. Tamir, S. Vardi, and Ning Xie. Fast Local Computation Algorithms. In *Proceedings 2nd Symposium on Innovations in Computer Science (ICS)*, pages 223–238, 2011.

[29] M. Szegedy and S. Vishwanathan. Locality based graph coloring. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, pages 201-207, 1993.

[30] J. Schneider, and R. Wattenhofer. A log-star distributed Maximal Independent Set algorithm for Growth Bounded Graphs. In *Proc. of the 27th ACM Symp. on Principles of Distributed Computing*, pages 35–44, 2008.

[31] J. Schneider, and R. Wattenhofer. A new technique for distributed symmetry breaking. In *Proc. of the 29th ACM Symp. on Principles of Distributed Computing*, pages 257-266, 2010.

[32] M. Wattenhofer and R. Wattenhofer. Distributed Weighted Matching. In *Proceedings 18th International Conference on Distributed Computing (DISC)*, pages 335–348, 2004.